Asian Journal of Information Technology 15 (1): 122-135, 2016
ISSN: 1682-3915
© Medwell Journals, 2016

QoS in Wireless Multimedia Sensor Networks by Contextual Routing with Octree Data Gathering

[1]G. Vithya and [2]B. Vinayagasundaram
[1]St. Joseph's Institute of Technology, 119 Chennai, Tamil Nadu, India
[2]Anna University, Chennai, India

Abstract: Wireless Multimedia sensor transmitted the data packet in real time heterogeneous networks are named as task driven data. The task driven is classified as event driven, time driven, query driven and normal data. To support QoS, the data packets are routed from the Cluster Head (CH) to the sink based on contextual priority of data. The collected packets from the Cluster Heads (CH) are routed and they are stored in an octree structure in the network gateway by the proposed algorithm, Contextual Routing with Octree Profiled data Gathering (CROPG). It has two phases, in Phase I, a Priority-in-Priority Routing (PPR) is developed to balance the load and delay by checking a priority between the cluster Heads. It is by analysis the packet type and packets are routed based on the traffic. In Phase II the packets are stored in octree structure to organize the contextual data. In this study, three novel features such as priority, resilient routing and octree data storage are looked upon as the central elements.

Key words: Contextual routing, priority-in-priority routing, octree gathering, data storage, central elements

INTRODUCTION

In a multipath environment as well as high resource constraints for a multimedia packet a call for an optimized contextual routing is in need. A context of data is viewed in terms of its task driven. In contextual routing the data are to be analyzed qualitatively by assigning priorities between the tasks and quantitatively by measuring the number of packets to be transmitted during the time period (t) with the minimum number of hops to reach the sink.

The task is classified as event driven, time driven, query driven and normal data. A catastrophic occurrence such as a fire breakout and other natural disasters are identified as an event and are said to be the event driven data, whereas periodical surveillance are labeled as Time driven. The queries made by the sink or the gateway node to various other nodes are as query driven data and surveillance data are called as normal or mundane data.

Phase I categorize the data as Event driven, time driven and assigns a priority between them. This phase, Priority-in-Priority Routing (PPR) also facilitates to choose an optimum or a feasible path to reach the sink. A novel priority model is proposed for assigning a high priority for event driven and a relatively lower priority for time driven data and for data organization. Real-time traffic poses a need for threshold to control the delay and jitter. Phase II is considered exclusively for octree data profiling.

The proposed CROPG is a real time protocol that supports QoS with the help of the significant features such as priority routing, load balancing, octree data gathering. It ensures balanced resource utilization by proper resource allocation. Overloading and intersection problems are overcome by the assigning priorities to the packets and choosing optimal or a feasible path for forwarding to avoid delay. Ultimately, in the sink the received packets are structured in an octree format.

Literature review: The routing in the WMS network by the factors such as QoS constraints, multimedia data handling, data delivery modes, network architecture are summarized (Ehsan and Hamdaoui, 2012) and there is only some protocol to support QoS by event driven. One of the QoS characteristics such as reliability of data model for Event driven data are discussed (Abazeed *et al*., 2013).

There are some other factors for routing and their challenges (Huang *et al*., 2014) are shown in Fig. 1 to support QoS routing in WMS network. The following review explains about the drawbacks of Qos challenges in previous researches.

For power consumption, by using cluster head architecture, higher probability to become a cluster head by residual energy (Li and Xu, 2011). In addition, a vice cluster head is added to the architecture to forward the data to support the load of the node energy by

Corresponding Author: G. Vithya, St. Joseph's Institute of Technology, 119 Chennai, Tamil Nadu, India

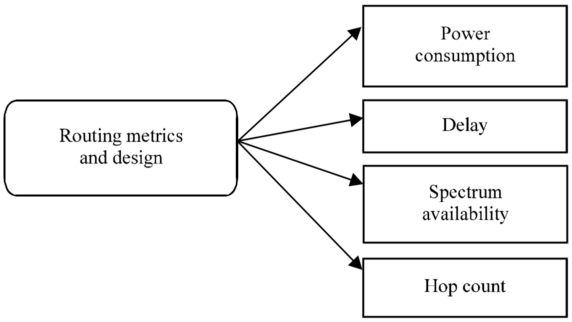

Fig. 1: Routing metrics and challenges

dissipation. The drawback of this algorithm is that the dissipation is not minimal and hence consume more energy indirectly.

For Delay, TEEN and APTEEN are used for episodic data gatherings and replying to time critical events technique to avoid delay only between node and cluster head but not to sink node.

With the spectrum avaliable to deliver data to a destination within a given delay can be by means of prioritized queue. The priority is determined by a function of traffic class, efficient prioritization and redundancy control (Lee *et al*., 2014). Classifies the traffic based on the applications such as to collect various form of sensing information and prioritize each type of data which is based on the request condition of the application service and provide differentiated data transmission only if the size of the network is small (Jabbar *et al*., 2015).

Furthermore, to support QoS routing along with residual energy, idle time, queue length and in addition to the measurement of links quality between nodes for selecting the most efficient and reliable paths to the destination (Machado *et al*., 2013). Routing by Energy and Link quality (REL) selects routes on the basis of end-to-end link quality estimator mechanism, residual energy and hop count for the efficient usage of the spectrum (Incebacak *et al*., 2013). For the hop count, by using Multipath routing for energy balancing which prolongs the network lifetime as compared to single-path routing (Jiang *et al*., 2013). But, there is no an optimal or a viable path hop to establish between the source and the destination upon selecting the CH and secondary CH. So, above all protocols are vulnerable because of no flexible method for forwarding priority data and to choose differentiated link quality for the path to balance the network life time.

Therefore, an energy efficient contextual routing protocol has to be designed in such a way that should satisfy the above constraints. And, by viewing and overwhelming problems in the existing works in the network and transport layer a small engineering work is required to support priority based Qos routing and maintains energy of the network by contextual data transmission. Efficient data aggregation still suffers by high redundant data transmission, efficient data organization (Shebaro *et al*., 2015) because of not using contextual access control with priority organisation (Hossain *et al*., 2013; Vithya and Vinayagasundaram, 2014).

Increase the metrics with efficient data aggregation, to eliminate the redundant data, determination of the routing paths for considering 100 times more in transition rate, low delay, traffic density and network lifetime is by Contextual Routing with Octree Profiled data Gathering (CROPG) Algorithm.

MATERIALS AND METHODS

The overview of Contextual Routing with octree Profiled Data Gathering (CROPG) is manifested as follows:

- Upon arrival of the packets from the source, they are routed to the sink based on the context. The contexts of the packets are identified by using task identification algorithm the packets are routed to the sink, where the sink acts as a gateway to other networks
- A clustered architecture assists when several events occur simultaneously. To curb the effects of delay and jitter by assigning priority between the tasks of neighborhood clusters and a Priority-in-Priority Routing (PPR) is used to calculate priority between the neighbor clusters
- To choose the intermediate nodes for routing and either to use optimal or feasible paths by using a Feasiopt algorithm which is used to balance the load in the network
- The data received from the network is organized in the sink as an octree structure. The search time and overhead are alleviated by Octree Profiled data Gathering (OPG) Algorithm.In this algorithm ,each node in the octree is considered as a cube to store and organise the contextual data such as:
  - Event driven packet
  - Time driven packet
  - Query driven packet
  - Normal driven packet

Figure 2 explains the overview of CROPG

Task identification algorith: Due to cruciality of data the task of the network is to be identified. A member node in

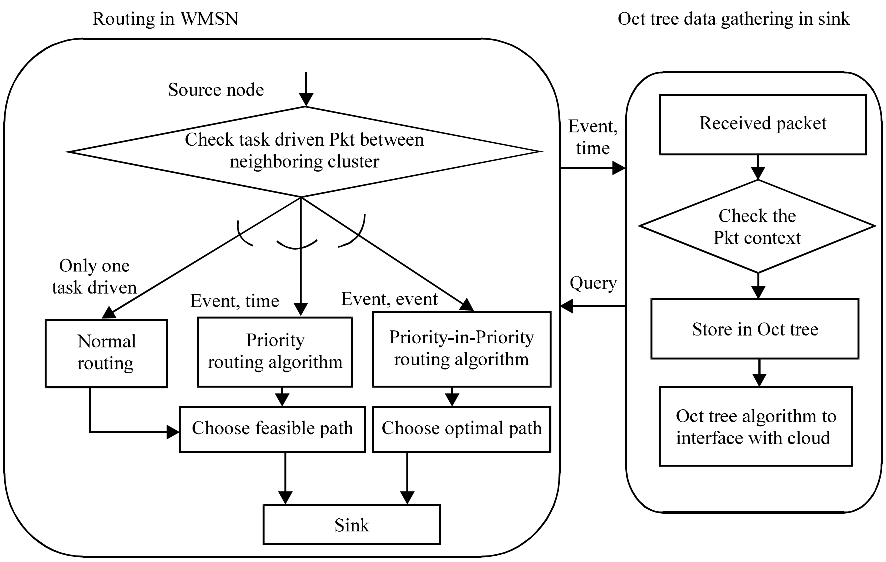

Fig. 2: Overview of CROPG

a cluster sends a beacon signal to a Cluster Head (CH) whenever an event occurs. In a heterogeneous clustered architecture, nodes with the highest level of attributes such as residual energy, success ratio and less time taken to deliver a packet are known to be cluster heads to route the event data. Nodes in the next order of importance in terms of attributes are called as proxy cluster heads to route the time data. Table 1 shows the temperature sensor value with its priority used to calculate priority. The Signal frame comprises of a unique wireless Node Identifier (SNI), time, Scalar value (Sv), Hard threshold (H_t), Soft threshold (S_t) and priority (pr).

The hard threshold represents a finite scalar value above which a node switches on its transmitter and sends a beacon to the cluster head. Soft threshold is introduced in order to increase the network lifetime. When the node reaches the hard threshold, the next transmission of a beacon signal occurs in the current cluster period, only if the present value of sensed temperature exceeds the hard threshold by an amount greater than or equal to soft threshold. Figure 3 shows the Task identification attributes of WS node.

In a Heterogeneous network, if non-CHs scalar sensor data are higher than a hard threshold, e.g., temperature >50°C for fire detection applications, this can be a possibility of an event occurrence. Such a beacon requires to be serviced immediately and this is depicted by setting the variable Pr to 1. Index variable i, allows to corroborate soft threshold condition. A WMS node in the cluster identifies the real impact of the incident, i.e., fire, in the environment and routes the

Table 1: Scalar value (Temperature)

Temp in Faren (F)	Scalar priority (Temperature)
>80°F	111
>60°F	110
>50°F	101
>40°F	011
<40°F	001

Node ID (SNI)	Time (T)	Scalar value (Sv)	Hard threshold (Ht)	Soft threshold S(t)	Priority (Pr)

Fig. 3: Task identification attributes in WS node

multimedia data to the sink. Based on the packet format the task is identified in a cluster by the below Task identification Algorithm A:

Algorithm A:

```
Task identification ()
  { If Sv(t) >= H_t
      CID. Pr=1; i=1
    Else
      CID.Pr=0; i=0
   If i=1
  If Sv (t) - H_t >= (S_t)
   CID.Pr=1;    }
```

So, the task such as event or time is identified based on the priority value in a cluster Head (CH). If Pr = 1 then an event is occurred. The packet in a CH routed to the sink in the following packet format.

Figure 4 explains the Packet Format in CH. At the same time, after identifying a task in a cluster, the network is analysed by checking if any event or time driven data occurred in the neighbour clusters. There may be a chance of event driven can occur in a cluster and

Time (T)	CH ID (CID)	Packet ID (PID)	Priority status (PID)

Fig. 4: Packet format in CH

neighbour cluster or event driven in one and time driven in neighbour cluster or time driven in both cluster and neighbour clusters. They are collectively solved by using priority-in-priority Routing algorithm which invokes routing such as normal, priority, priority-in-priority Task to route the packet to the sink.

Priority-in-Priority Routing (PPR): Priority is assigned between event driven and time driven data by employing the Task Identification Algorithm. Based on the Task Priority algorithm, routing type is established.The PPR algorithm in Phase I invoke the following routings as follows:

- Priority Task algorithm () is used if events in one cluster and time driven in neighboring clusters. Event driven data uses the cluster head and the time driven, uses proxy cluster head for transmission.
- Priority-in-Priority Task algorithm (PPR ()): If two events occur simultaneously between the neighboring clusters then by the algorithm PPR (). In PPR(), The cluster head selects an optimal path for forwarding by comparing the priority values between the CH and other CH to choose the viable path. It helps the routing protocol to restrain packet loss as well as ensures the transfer of all the crucial data to one or more destination
- Normal task () is used if time driven occurred both in a cluster and neighboring clusters

Figure 5 explains the various types of routing in Priority -in-priority routing (Alogrithm B).

Alogrithm B:

```
Priority Task Algorithm ()            //Calculate priority between neighbor
clusters
     {  For all cluster do
          {
          Task identification ();
          If (CID .Pr = 1 ) && (NrCID .Pr=1) then
         Priority-in-priority Task ()
       Else
           If (CID.Pr = 0) && (NrCID .Pr=1) || (CID.Pr = 1)  && (NrCID
.Pr=0) then
         Priority Task ()
     Else
        Normal Task ();          }
     feasiopt algorithm ();//Route selection  for PPR
     OPG algorithm ();// Octree profiled data Gathering            }
```

The following study explain the priority routing and priority-in-priority routing depends on event occurs in the neighboring clusters.

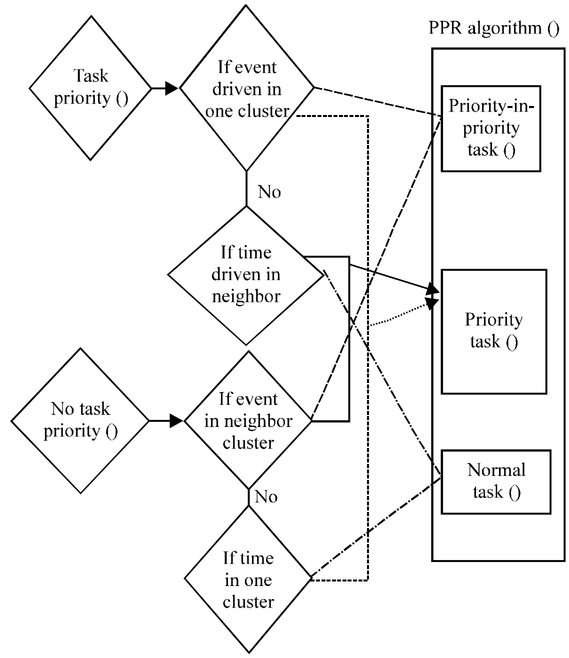

Fig. 5: Sceleton of priority-in-priority routing

Priority driven task: A widely spread hierarchical sensor network poses co-ordination between cluster heads as a stipulation. Behavioral model aids in the execution of either priority driven or priority-in-priority algorithm. Instances in which one Cluster Head (CH) is involved in Event driven operation and the other in Time driven simultaneously, requires priority to be set for routing data to the sink. The process involved is termed as a priority driven algorithm. For example, an event is identified by cluster head1 and the neighborhood CH is involved in surveillance at an arbitrary time t1, then priority assigned to the event driven packet (Alogrithm C).

Alogrithm C:

```
Priority Task ()
   { If (CID .Pr = 1) then
     Choose CH as source cluster
      else
     Choose NrCID as source cluster  }
```

Priority-in-priority driven task: In case of two events taking place concomitantly, data forwarding is based on the priority value set for each of the events. Priority is hinged on the scalar value and other attributes pertaining to scenarios such as the attribute population. For instance, by comparing a bomb blast that occurs in less populated areas and at the same moment as a fire explosion in a fuel carrying vehicle in densely populated area, then it demands a higher priority for data forwarding. The scantling values are in the following Table 2 and 3. Table 2 shows the population priority, Table 3 shows Prioriy-in-priority values.

Table 2: Population priority

Total Population Count ($TPC_{1=1000}$)	Pop-priority
>1000/cell	111
>800/cell	110
>600/cell	101
>400/cell	011
<100/cell	001

Table 3: Priority-in-priority

Clusters	Scalar value (Temp.) (Sv)	Pop. count (poc)	Time (T sec)	Priority status (prs)
Cluster 1	111	111	10.00	P1
Cluster 2	111	111	10.05	P2
Cluster 3	-	101	-	-
Cluster 4	-	011	-	-
Cluster 5	-	111	-	-

For comparisons, the measured temperature is taken as the scalar value and the other attribute considered is the population affected which can be acquired with the aid of Global Positioning System (GPS). Priority status is calculated based on scalar, population priority with time factor (Alogrithm D).

Alogrithm D:

```
Algorithm; Priority-in-Priority Routing-PPR ()
   PPR()
    {Let Sv(p) as Scalar priority;
      Let  Poc as population count;
         If (CID1.Sv(p)> NrCID. Sv(p) ) || (CID1.Sv(p) < = NrCID. Sv(p))
Then
          If  CID1. Poc > NrCID.poc then
             CID1.prs= p1
         If p1
            CID1 as source cluster
          Else
            CID1.prs= p2
            NrCID as source cluster;
           }
 If time driven data is in a CH and neighbour CH then normal routing taking
place.
Normal Task ()
   {Choose CH as source cluster}
```

Feasiopt algorithm: By default, all three types of routing use path hop algorithm to select a next hop. A path prediction model is developed by constructing a Feasiopt algorithm (Fa) which is based on the attributes of the cluster heads and proxy cluster heads. In algorithm fa() routing is enforced by establishing residual energy of the node, packet forwarding rate of the node, success ratio of the node to differentiate an optimal path or a viable path. During the time driven the node starts transmitting their sensor readings by selecting a viable path to avoid congestion. Furthermore, priority driven and priority-in-priority algorithm depends mainly on feasiopt algorithm in order to carefully chooses the intermediate nodes for routing and differentiates between optimal and feasible paths. Path prediction based on the behavior of a wireless sensor node can be characterized through its attributes such as:

- Residual energy of the node
- Packet forwarding rate of the node
- Success Ratio of the node

Residual energy of the node (α_i): A residual value is calculated for each node based on Packet Sending/Receiving Rate (PSR/PRR), number of packets sent or received over a period of time. A node serves either as a cluster head or a proxy cluster head depending upon its residual energy:

- E Trans = P Trans×t Trans
- E rec = P rec×t rec
- E idle = P idle×t idle
- E sleep = P sleep×t sleep

Where (E trans, E rec, E idle and E sleep) are the consumed energy, (P trans, P rec, P idle and P sleep) are the circuitry power consumption, (t trans, t rec, t idle and t sleep) are the periods of time spent by the node, for each state (transmit, receive, idle and sleep), respectively.

Accordingly, the total energy consumption (E total) for a node to transmit and receive a packet can be calculated as defined in Eq. 1 (Abazeed *et al*., 2013):

$$E\ total = E\ Trans + E\ rec + E\ idle \tag{1}$$

Packet Forwarding Rate of a Node (PFR_N): The Packet Forwarding Rate (PFR) of the node is the ratio between the number of signals with Received Signal Strength (RSS) exceeding the threshold and Packet arrival process. The RSS is the Measurement of power present in received radio signals and Packet Arrival Process (PAP) is a Poisson process in which the inter arrival time distribution is exponential with rate λ:

$$P\,(n\ arrivals\ in\ interval\ T) = \frac{(\lambda T)^n e^{-\lambda T}}{n!} \tag{2}$$

The above factors decide the number of packets that the node receives from its neighbors and consequently forwards to its next node within a period of time.

Success Ratio (SR): Success ratio of the node is defined as the ratio of Packet forwarding rate of the node to Packet Dropping rate. Packet Dropping Rate (PDR) is the number of packets that were sent to a certain node but were not forwarded by that node:

$$SR = \frac{PFR_N}{PDR} \tag{3}$$

Apart from the above three factors, packet forwarding rate of a link plays a crucial role in path prediction and forwarding.

Packet Forwarding Rate of a Link (PFR_L): The general format for the packets forwarded from one cluster head to next cluster head is based on:

$$\delta n\,(i \rightarrow i+1) = \frac{LQ}{dis} \qquad \text{where } i = 1...p \tag{4}$$

Where:

i = Index variable
p = Number of cluster heads
dis = Distance between two cluster heads
LQ = Link quality

Considering two cluster heads CH1 and CH2, LQ is decided by the Latency (L_T). The loss rate LR for the link L is calculated as:

$$LR = 1 - \frac{D_{re}}{D_{se}} \tag{5}$$

Where:

D_{re} = Number of data packets received
D_{se} = The number of data packets sent over L

$$L = \begin{cases} \text{Delay tolerant} & L_T > ThLT \\ \text{Delay intolerant} & L_T < ThLT \end{cases} \tag{6}$$

$$L = \begin{cases} \text{Loss tolerant} & LR > ThLR \\ \text{Loss intolerant} & LR < ThLR \end{cases} \tag{7}$$

From Eq. 1 and 2, Normal link (C_L) = $L_T > ThL_T$ and $LR > ThLR$. Critical link (N_L) = $L_T < ThL_T$ and $LR < ThLR$. Critical link is a link that is delay and loss intolerant whereas normal link is both delay and loss tolerant. The feasiopt algorithm is described in Alogrithm E:

Alogrithm E:

```
Algorithm: feasiopt algorithm() //Route selection for PPR
For each next hop from source to destination do
feasiopt algorithm()
    Let Rv_a, Rv_b = neighbors to source
    Let Pf_a,Pf_b = Packet forwarding rate of neighbor nodes from source
    Let SR_a,SR_b= Success ratio of nodes
    Let LT_a - Latency from source to
     If((Rv_a>=Rv_b)&(Pf_a>= Pf_b)&(Sr_a>= Sr_b))
     Nexthop = a
    Else if ((Rv_a<=Rv_b)&(Pf_a>= Pf_b)& (Sr_a>= Sr_b)) || Rv_a>=Rv_b) &(Pf_a<=
Pf_b) & (Sr_a>= Sr_b))|| Rv_a>=Rv_b) &(Pf_a>= Pf_b) & (Sr_a<= Sr_b))
     A_1= Rv_a+Pf_a+ Sf_a/LT_a
     B_1= Rv_b+Pf_b+ Sf_b/LTb
     If A_1>B1
     Nexthop a
     else Nexthop = b;
     endif
    Else nexthop = b
    End for
```

Similar to existing reactive routing protocols, whenever a route to a destination node is required, a route discovery process is initiated in order to locate the destination. To do so, a route request packet is forwarded to the immediate neighbors that, in turn, forward the route request to their neighbors and so on until the destination is reached. The next hop selection is based on the attributes of the nodes and links as already explained. Since, the packet arrival pattern is not predictable in case of event driven routing and the capacity of each node also differs in the WMS network, lifetime of a network becomes tedious to maintain. feasiopt algorithm aids in balancing the load and hence improves lifetime and performance of the network.

Optimal path and feasible path: An optimal path is employed only with cluster heads in order to increase the speed of the routing due to the event criticality. In a feasible path, the proxy cluster heads are involved in routing time driven packets to balance the load. End-to-end Latency (EL_T) plays a paramount role in deciding the performance measures for both optimal and viable paths. EL_T of a path T is formed by the composition of the delays of its intermediate links. Let $L_T(a, b)$ be the delays of intermediate links with a and b denoting the start and the end of the link respectively. Then,

$$EL_T = \sum_{\forall (a,b) \in T} L_T(a,b)$$

An optimal path is composed of a greater number of critical links that are loss and delay intolerant whereas a viable path is composed mainly of normal links and are found to be loss and delay tolerant.

Octree profiled data Gathering (OPG PhaseII): In the Sink, the trade-off between minimal storage space and minimal processing time are managed by gathering and organize the data in an octree structure. In phase II of CROPG, the organization of data at the sink takes the form of an octree, which are largely employed to provide resilience for wireless losses. At the apical level, the octree structure stores the whole environment data packet.

Being a recurrent structure, each node in an octree represents a cube and the subsidiaries represent finer details of the cube. Consummation of an octree structure relies on effective storage that accelerates search time. Effective storage is executed by the estimation of data exploiting context priority which defines the service order as event, time, location, query and normal. The contextual model aims to interface effectively with the sink by autonomous data gathering. Figure 6 and 7 shows the

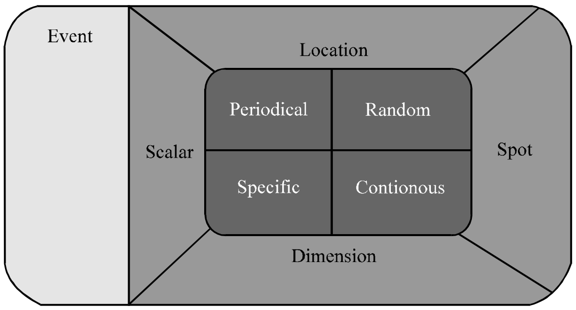

Fig. 6: Skeleton of cube

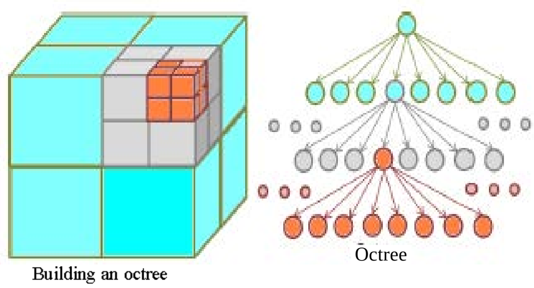

Fig. 7: Event contextual cube

sensed image primitives by the contextual data to reach to the sink. An overview of octree profiled data Gathering is explained as follows.

Octree Profiled data Gathering (OPG) in CROPG (PhaseII): The sink receives the packets from eight different directional wireless multimedia sensor networks and they are stored in octree format. Each node in octree is organised as a cube. For effective storage, the space in the sink is divided recursively into cubes. Then eight directional network data are stored in the form of a cube structure.

The contexts are analysed and store the data in the form of quad tree. Each cube is divided into three quad trees to store task, spatial, temporal data.

Contextual data such as event, time, query and normal data are stored in a quad tree called a Task quad tree (Front view).

To store spatial data such as location, spot, dimension and scalar data are stored in the spatial quad tree. Spatial data provide detailed information about the physical parameters associated with the event, for instance the exact place of occurrence of a fire accident as well as the distance of spread and its core temperature (Side view).

To store temporal data namely periodical, random, specific and continuous data are stored in temporal quad

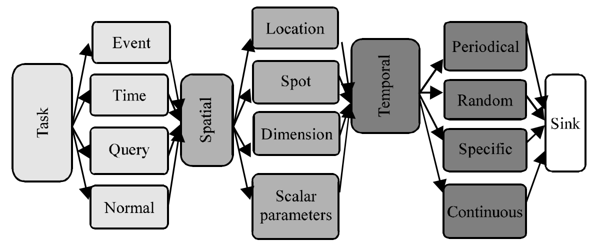

Fig. 8: Contextual estimation of data packet in the sink

tree (Top view). A block of data stored together is called chunks which contains data of 3 quad trees. The sink with unperpin of the context estimator to identify the context of the packet and by using the tagging attribute to measure the occupied status. To store data in the cube by the following steps.

Cube estimator: If priority status (prs) has any value, then it is considered as event data. If pr = 0 then it is considered as time driven data which is by comparing the Time (T) in the packet format with the surveillance slotted time. If the time matches then it is considered as Time driven. Otherwise, it is considered as normal driven data. For query driven data is identified by Req and Reply Message. Figure 8 shows the contextual estimation of data Packets and each node in an octree describes a Multimedia Sensor network data and the sub-tree rooted at that node represents the temporal and spatial data.

Cube analyser: The quad chunks are operated by the index tagging attributes. They are as Top of storage status pointer-Tc, Bottom of storage status pointer-Bc, current status pointer-Csp.

Octree Profiled data Gathering (OPG) algorithm: A cube is divided into 4 chunks. The task Qquad tree such as event, time, query and normal data packets are stored in chunk0, chunk1, chunk2, chunk3, respectively. In a chunk the spatial and temporal data packets are linked by using hash indexed link list format. The occupation level is maintained by incrementing tagging attributes. Input: CID, NrCID, Time (T), Priority status (pr), PID. Output: cube in octree.

OPG Algorithm ()

```
{struct hash *octtable = NULL
struct spatial
{Char**loc[50]; Char**spot[50];Char**dim[50];Char**scal[50];
} quadspatial;
struct Temporal
{Char**periodical[50];          Char**random[50];Char**specific[50];
Char**contin[50];
} quadtemporal;
struct node
{int priority; struct quadtask,struct quadspatial;struct quadtemporal;struct
node *next
```

```
}
struct hash
{Struct node *head;
Int Tc, Bc, Csp=0;  };   // to identify storage level
Starting of the program
  octreeSink ()
{ get CID                              //get CH Id value
   for i= 1 to 8
   {   if CID = =Cube[i]           // for eight direction network by 8 cube i=
1...8
                    insertTo  cubeHash(int  priority,  struct  quadspatial,  struct
quadtemporal)    }}
{struct node * createNode(int priority, struct quadspatial,structquadtemporal)
{    struct node *newnode;
   newnode = (struct node *) malloc(sizeof(struct node));
   newnode->priority = CID.Pr;
   strcpy(newnode->spatial.loc, CID.GPS.loc);
   strcpy(newnode->spatial.loc, CID.GPS.spot);
   strcpy(newnode->spatial.loc, CID.GPS.dim);
   strcpy(newnode->spatial.loc, CID.GPS.scal);
   newnode->next = NULL;
   return newnode;}}
void insertTo cubeHash(int priority, struct quadspatial, struct quadtemporal)
{
  {   If CID.pr >=1
       {strcpy  (chunk[0] , quadTask[event data) ;                       // Store
Event data in Chunk [0]
      quadtemporal= random;
      int cubendex = chunk[0];}
       else
          If (CID.pr <=1) && (CID.T == sink.T)
            {strcpy (chunk [1], quadTask[time data];                // Store Time
data in Chunk [1]
            quadtemporal= time;
           int cubendex = chunk[1];}
      else
            {strcpy(chunk [2],Task[normal data]; }               // Store Normal
data in Chunk [2]
            quadtemporal= contious ;
            int cubendex = chunk[2];}
       If (sink.reqmsg) && (sink.repmsg)
           { strcpy (chunk [3],Task[quary data];                    // Store Query
data in Chunk [3]
            quadtemporal= specific;
            int cubendex = chunk[4]}
            }
               struct   node   *newnode   =   createNode(int   priority,   struct
quadspatial,structquadtemporal)
     }
   /* head of list for the chunk with index "hashIndex" */
   if (!octtable[cubeindex].head) {
      octtable [cubeindex].head = newnode;
      octtable [cubeindex].tc = 1;
      return;    }
   newnode->next = (octtable[cubeindex].head);    /* adding new node to the
list */
     octtable [cubeindex].head = newnode;                /* update the head of
the list   */
   octtable [cubeindex].tc++;
   return;   }
   Csp= octtable[cubeindex];}
```

RESULTS AND DISCUSSION

PPR in CROPG algorithm (Phase I): The most important metrics that are measured in the following experimental study with regard to the QoS in PPR of CROPG algorithm. Figure 9 shows CH and proxy cluster head selection. Figure 10 shows only time driven occurred in WMSnetwork.

Figure 11 shows only event driven occured in WMS network. Figure 12 shows event in one cluster and time driven in neighbor clusters. Figure 13 shows the time driven in cluster and neighbor clusters.

Figure 14 transmission of event in cluster and in neighbor clusters using PPR algorithm.

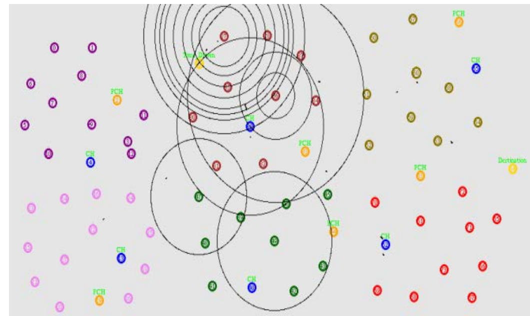

Fig. 9: CH and proxy cluster head selection

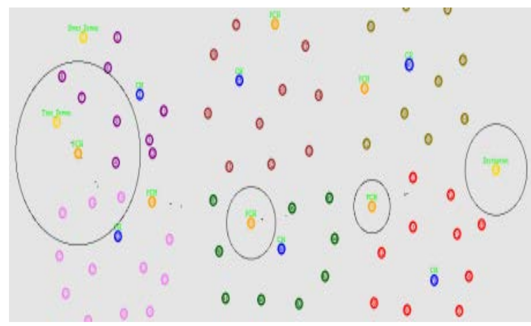

Fig. 10: Only time driven in WMS network

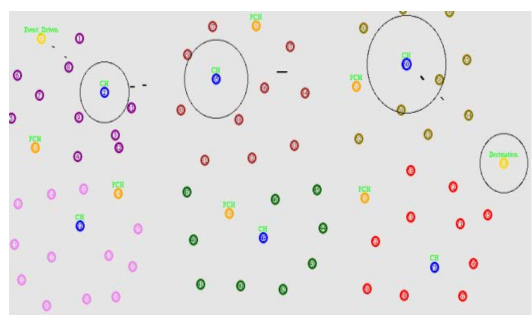

Fig. 11: Only event driven in WMS network

Parameters for PPR in CROPG (Phase I): The NS2 Simulator is used to evaluate the PPR protocol

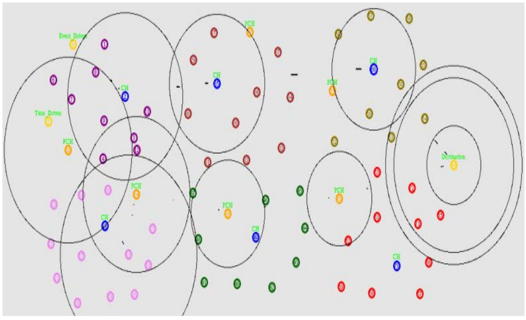

Fig. 12: Event in one cluster and time driven

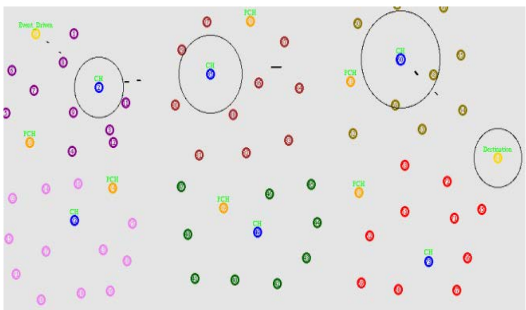

Fig. 13: Time driven in cluster and neighbor clusters

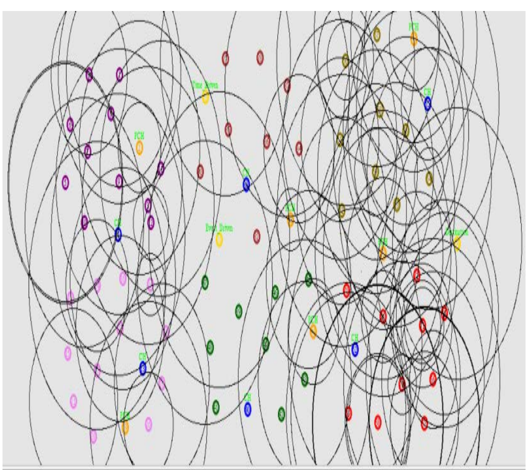

Fig. 14: Shows the transmission of event in cluster and in neighbor clusters using PPR algorithm

performance. Table 4 displays the parameters used in the simulation. Each node generates a 512 byte data packet for every second. An initial energy of 10.1 J is given for every sensor node. The power consumption when the sensor nodes in transmit mode, receive mode and idle mode are 0.660, 0.395, 0.035W, respectively.

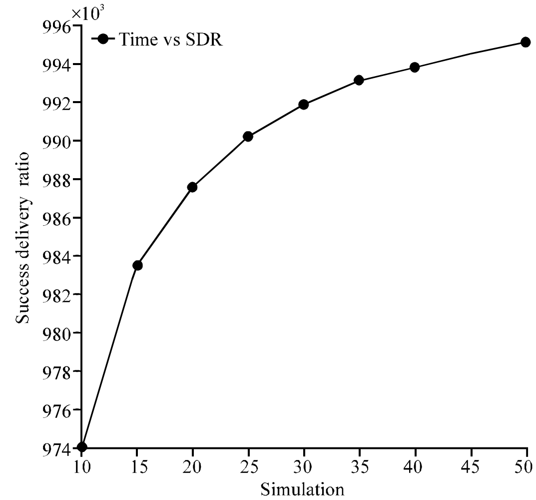

Fig. 15: Sccess delivery ratio

Table 4: Simulation parameters

Simulation parameters	Values
Simulation time	50 sec
Traffic	CBR
Packet size	512 Bytes
Routing protocol	AODV with PPR()

The metrics used to assess the efficiency of the PPR are Success Delivery Ratio (SDR) Energy cosumption, residual energy and end-to-end delay.

Success delivery ratio: Success ratio of the node is defined as the ratio of Packet forwarding rate of the node to Packet Dropping rate. It is calculated from Eq. 3-7 and the result shows in Fig. 15.

Energy consumption: The total energy consumption (E total) for a node to transmit and receive a packet can be calculated as defined in Eq. 1 as E total = E Trans+E rec+E idle and it shows in Fig. 16.

Residual energy: A residual value is calculated for each node based on Packet Sending/Receiving Rate (PSR/PRR). Available energy after the number of packets sent or received over a period of time. It shows in Fig. 17.

End-to-End Delay is nothing but the time taken by the packets to reach the destination or sink node. Figure 18 shows the end-to-end delay of PPR protocols under varying arrival rates by using feasiopt algorithm.

We compare the performance of PPR with ICACR algorithm by the parameters such as Success Delivery Ratio (SDR), Energy cosumption, Residual Energy and end-to-end delay. ICACR formulates a generalized QoS-aware Routing Model on the basis of multiple

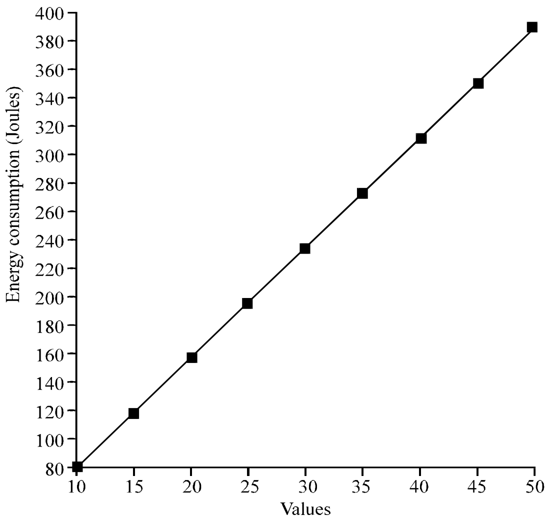

Fig. 16: Energy consumption

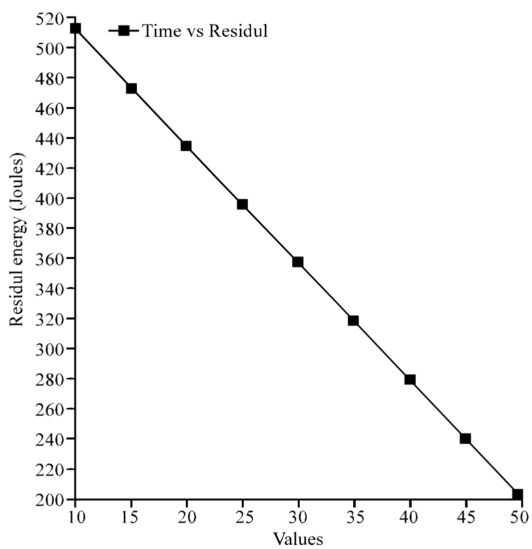

Fig. 17: Residual energy

routing metrics and priorities of packets. A 2D Plain-Based Routing algorithm IPACR which improves the standard ant colony algorithm by a clustering-based routing algorithm ICACR. Table 5 and 6 shows the SDR, energy consumption, residual energy, delay for PPR, ICACR algorithm, respectively. In this study, we evaluate the clustering-based ICACR with PPR from four aspects:

- The comparison of success Delivery Ratio (SDR) between PPR and ICACR
- Energy consumption, the one of the evaluation of routing metrics, including Transmit, receving idle time of a node using scalar data packets in PPR with ICACR

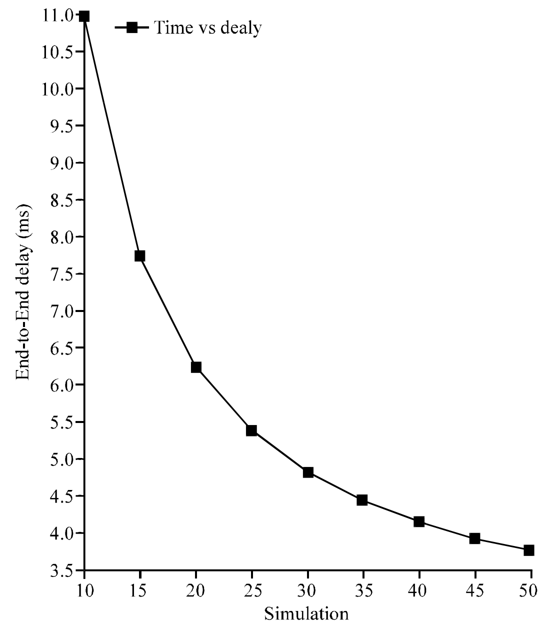

Fig. 18: End-to-End Delay

Table 5: Parameters of PPR algorithm

Time (t)	SDR value	Energy consumption	Residual energy	Delay
10	0.9741	79.2807	512.719	11.00200
15	0.9835	118.1570	473.843	7.74110
20	0.9876	157.0520	434.948	6.25144
25	0.9902	195.9390	396.061	5.39075
30	0.9919	234.7970	357.203	4.83131

Table 6: Parameters of ICACR algorithm

Time (t)	SDR value	Energy consumption	Residual energy	Delay
10	0.8835	109.558	482.442	257.638
15	0.8965	153.939	438.061	178.133
20	0.9233	154.274	407.726	131.996
25	0.9391	214.606	377.394	105.951
30	0.9495	244.940	347.06	89.262

- Residual energy, a real video quality evaluation between PPR and ICACR while considering the multi-priority of the video frame packets through multi-path
- End-to-End Delay includes packet loss rate, throughput and networklifetime of PPR with ICACR

The PPR protocol uses the feasiopt algorithm for choosing the best link. Figure 19 shows the SDR in PPR with ICACR. The PPR protocol provides priority routing with dynamic path which is based on the task driven. Figure 20 shows the influences in the improvemeny of the energy consumption of the sensor nodes compared to ICACR.

Figure 21 shows the Residual Energy of the nodes in the network using PPR and ICACR. Figure 22 shows

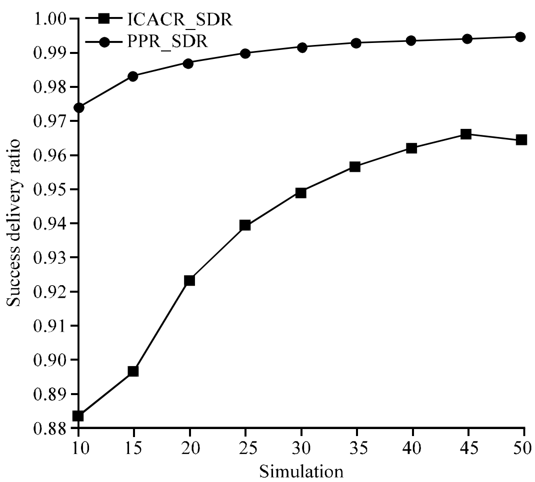

Fig. 19: Comparison of Sucesses Delivery ratio

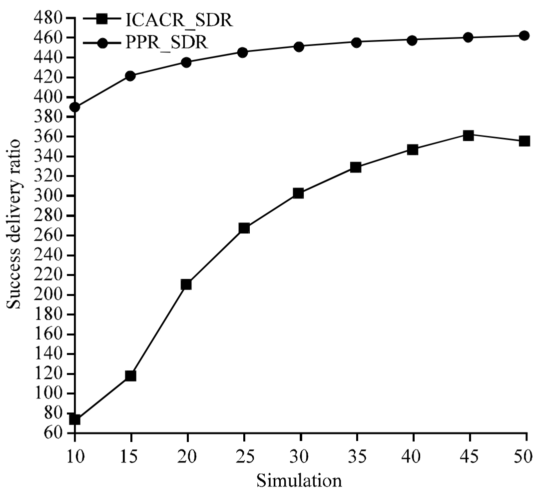

Fig. 20: Comparison of energy consumption between between PPR , ICACR

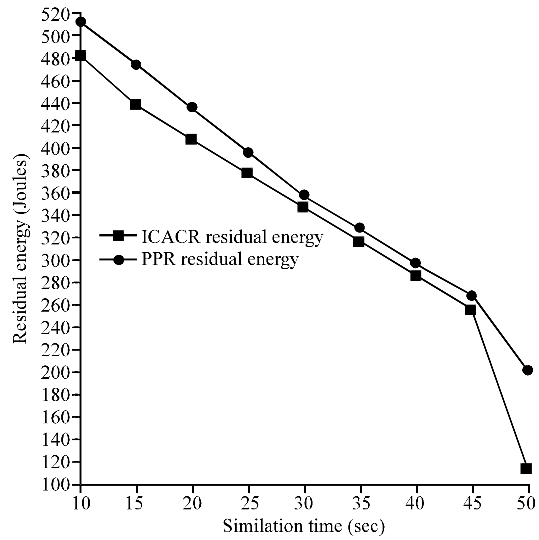

Fig. 21: Comparison of Residual Energy between between PPR , ICACR

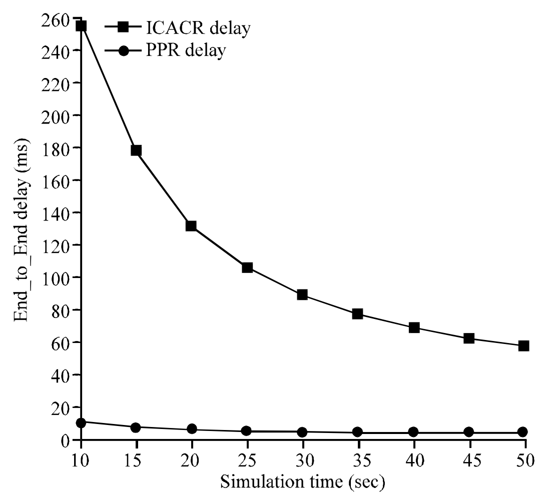

Fig. 22: Comparison of end-to-end delay between PPR and PPR and ICACR

that the delay of the PPR is lower than that of ICACR. Sub-sequently, the PPR follows minimum delay in identifying its neighbor node based on the priority value and thereby the transmission delay is reduced.

Parameters for Octree Profiled Data Gathering in CROPG (Phase II): The management of the octree data structure generally exhibits the O(log N) computational complexity which becomes nearly constant for a reasonable octree depth.

Evaluation metrics: In Fig. 23 evaluation metrices of CROPG. The detailed tree structure shows the vital few attributes that contribute to establishing QOS in wireless multimedia sensor networks. This study supports 3 types of task such as event, time and query driven data. Event driven with the threshold temperature and trained images. Same network can manage different types of task. The priority assigned between the neighboring clusters based on their criticality. If Event occurred in one cluster and time driven data in next cluster, then priority is assigned to event driven data. If events have occurred in both the clusters then priority-in-priority algorithm is used to identify the most priority.

Because of using priority the crucial packet gets a chance of routing first without delay. All nodes in the network are getting a chance of participation in routing to avoid overwhelming the same or particular node. Feasible or optimal routing can be used to reach the sink node.

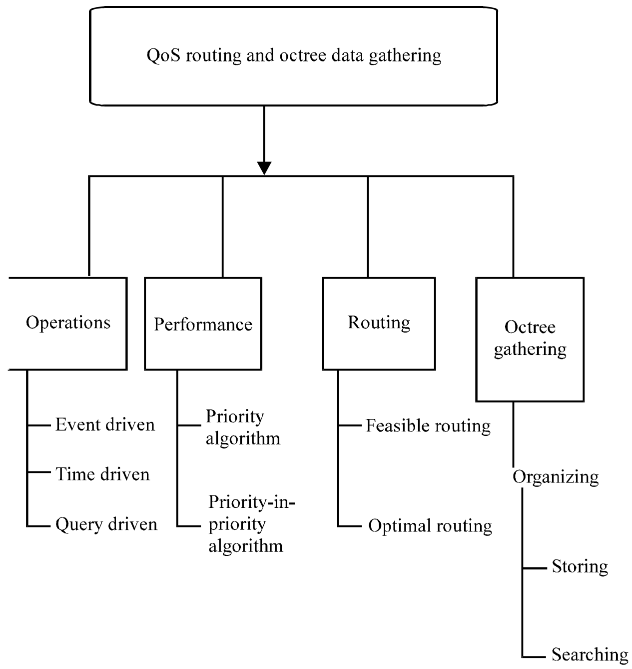

Fig. 23: Evaluation metrices of CROPG

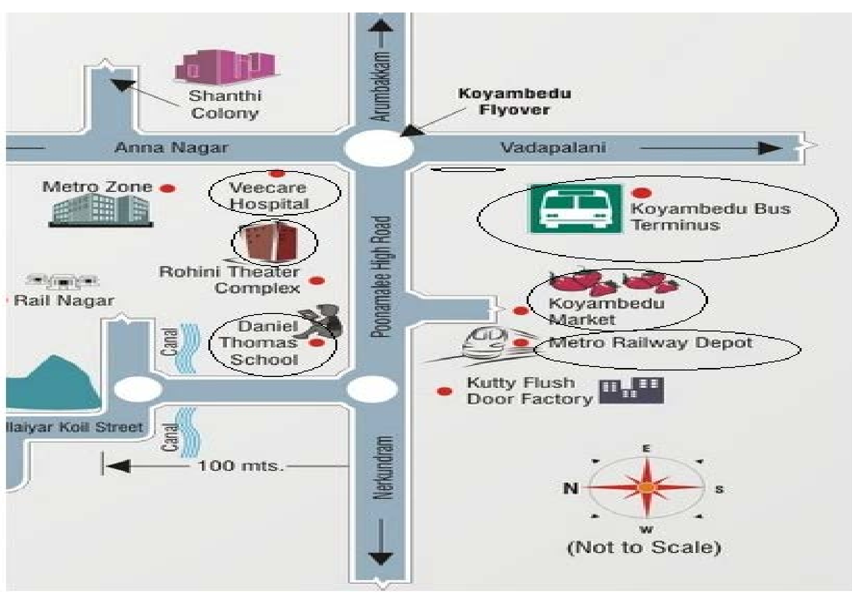

Fig. 24: Location of Koyembedu, Chennai, India

The task and their related information are stored in the sink in octree pattern, which is used to store and searching the packet in no time. The Qos is supported in the task, priority, routing, storing. Efficiency is the result of maximizing productivity with minimum effort or expense.

Priority Packet flow resilient routing (PPR) and Octree Profiled data Gathering (OPG) algorithm in CROPG (Phase I and II).

Figure 24 shows within a stretch of 2 miles in the capital city of Tamil Nadu in India, the population is irrationally high due to the presence of several prime

Fig. 25: Scenario in Koyembedu, Chennai, India

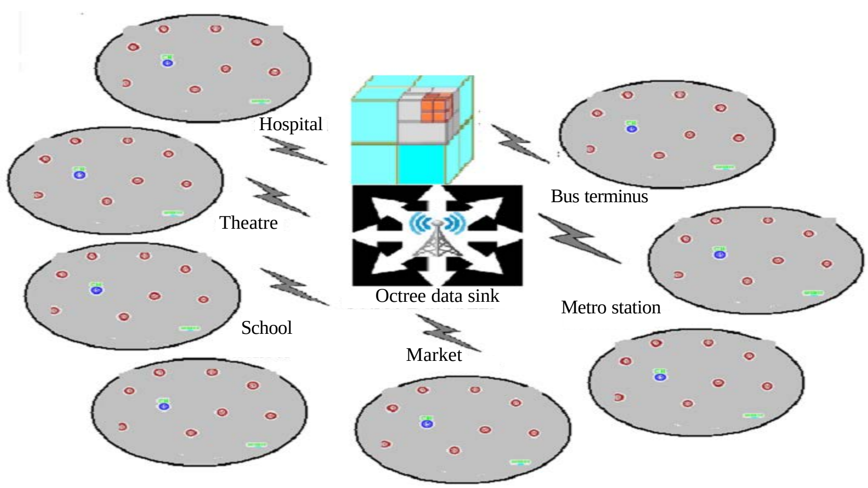

Fig. 26: Deployment of WMS Network in Koyembedu, Chennai, India

spots such as Bus terminuses, schools, hospitals and shopping complexes. Figure 25 shows the Scenario in Koyembedu, Chennai, India which houses Asia's largest Bus terminus as well as other crowd pullers such as theatre complex, metro station and a wholesale market, any anomaly that occurs within this stretch will potentially generate a humungous loss to the economy. For instance, a fire breakout that occurs in one of the places with >1000 people per square mile will cause a great loss of lives and property. In order to take timely action, wireless sensor networks with priority based routing is essential. The priority here is set based on the Total Population Count (TPC) as well as the distance (range, R_d) to which fire has spread. The severity of the fire is decided based on the temperature change in the environment (T_C). Figure 26 shows the deployment of WMS Network in Koyembedu and store the data in sink by octree structure.

CONCLUSION

In this study, a new contextual routing for WMS network called PPR is proposed which check the context and assign priority for the task. In order to handle two adjacent clusters with the same priority then a new PPR is proposed. From the results, it is strongly proven that PPR delivers maximum data with more success delivery rate and minimum energy consumption. To achieve network lifetime maximization and to balance the load, the feasiopt algorithm is applied to choose the next hop. It is clearly

Table 7: Priority table in WMS network

Temp. ranges (m)	Temp. in Faren (F)	Temp. priority
100-200	>80°F	111
50-100	>60°F	110
10-50	>50°F	101
1-2	>40°F	011
<0.01	<40°F	001

evident that the end to end packet delivery is greatly reduced by letting only the feasible path for more priority task and an optimal path for the lesser priority. The simulation results also prove that the PPR provides a less delay and high packet delivery ratio, compared to the existing ICACR.

A series of simulative experiments have been conducted to evaluate the proposed routing algorithm. The results show that the PPR is faster than the ICACR when the network scale becomes larger. The PPR outperforms the ICACR in all routing performance metrics. Again, foreshorten in finding the priority with the following Table 7 in futurework.

In Phase II in the Sink, minimal storage space and minimal processing time are managed by gathering and organize the data in an octree structure. But if more than three neighbor cluster node gets the same priority, then PPR is not obvious. Meanwhile, more effective octree profiled data gathering would be emphasized to act with cloud in the future research.

REFERENCES

Abazeed, M., N. Faisal, S. Zubair and A. Ali, 2013. Routing protocols for wireless multimedia sensor network: A survey. J. Sensors, 10.1155/2013/469824

Ehsan, S. and B. Hamdaoui, 2012. A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks. IEEE Commun. Surv. Tutorials, 14: 265-278.

Hossain, M.A., A.A.N. Shirehjini, A.S. Alghamdi and A. El Saddik, 2013. Adaptive interaction support in ambient-aware environments based on quality of context information. Multimedia Tools Appl., 67: 409-432.

Huang, H., X. Cao, R. Wang and Y. Wen, 2014. A QoS-aware routing algorithm based on ant-cluster in wireless multimedia sensor networks. Sci. China Inf. Sci., 57: 1-16.

Incebacak, D., B. Tavli, K. Bicakci and A.A. Kayhan, 2013. Optimal number of routing paths in multi-path routing to minimize energy consumption in wireless sensor networks. EURASIP J. Wireless Commun. Networking, 1: 1-10.

Jabbar, W.A., M. Ismail and R. Nordin, 2015. Multi-criteria based multipath OLSR for battery and queue-aware routing in multi-hop ad hoc wireless networks. Wirel. Netw., 21: 1309-1326.

Jiang, Z.Y., M.G. Liang and J.J. Wu, 2013. Dynamic source routing strategy for two-level flows on scale-free networks. PloS ONE., Vol. 8, 10.1371/journal.pone. 0082162.

Lee, S.K., J.G. Koh and C.R. Jung, 2014. An energy-efficient QoS-aware routing algorithm for wireless multimedia sensor networks. Int. J. Multimedia Ubiquitous Eng., 9: 245-252.

Li, F. and T. Xu, 2011. Energy-aware data gathering and routing protocol based on double cluster-heads. Proceeding of the International Conference on Consumer Electronics, Communications and Networks (CECNet), 2011, April 16-18, 2011, IEEE, XianNing, China, pp: 4059-4063.

Machado, K., D. Rosario, E. Cerqueira, A.A. Loureiro and A. Neto *et al*., 2013. A routing protocol based on energy and link quality for internet of things applications. Sensors, 13: 1942-1964.

Shebaro, B., O. Oluwatimi and E. Bertino, 2015. Context-based access control systems for mobile devices. Dependable Secure Comput. IEEE. Trans., 12: 150-163.

Vithya, G. and B. Vinayagasundaram, 2014. QOS by priority routing in internet of things. Res. J. Appl. Sci. Eng. Technol., 8: 2154-2160.